## PROGRAM

## //Program to create a simple class in a package and access it by another classs

E:\Myjava\Assignment\pkg

package pkg;

public class simple

{

public void show()

{

System.out.println("\t\tThis is a simple class.");

System.out.println("This is the my package.");

}

}

E:\Myprogram

import pkg.\*;

class importing

{

public static void main(String args[])

{

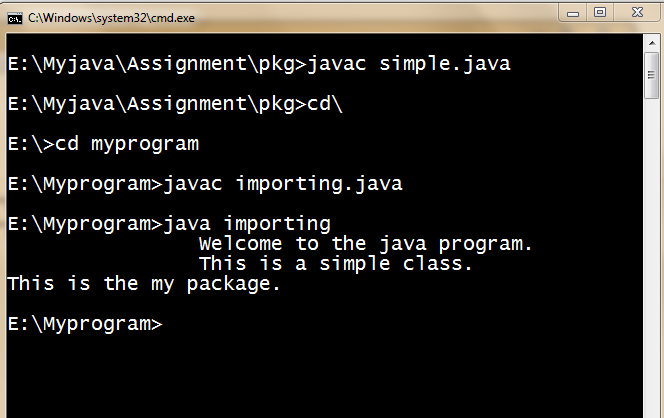
System.out.println("\t\tWelcome to the java program.");

simple ob=new simple();

ob.show();

} }

## OUTPUT:-



## PROGRAM

## //Program to create a String and perform all its operation upon that in java.

class str

{

String s="Dinkar Prasad Maury";

static void p(String s)//called when class is first time loaded.

{

System.out.println(s);

}

str()//constructor.

{

char Chars[]={'a','t','u','l','d','i','n','k','a','r'};

String s=new String(Chars,4,6);

p("Created string="+s);

String s1=new String(s);

p("Created from above string s1="+s1);

byte ascii[]={97,98,99,100,65,66,67,68};

String s2=new String(ascii);

p("Byte string="+s2);

String s3="Dinkar Prasad Maurya.";

p("Initialization by other method.s3= "+s3);

}

public void operation()

{

String s1="Atul Singh";

String s2="atul singh";

p("The length of the string s="+s.length()+"Bytes");

p("Concatenation:"+(s+s1));

p("equals():"+s.equals(s1));

p("equalsIgnoreCase():"+s1.equalsIgnoreCase(s2));

p("equals():"+s1.equals(s2));

p("startsWith():"+s.startsWith("Prasad",7));

p("endstWith():"+s.endsWith("Maury"));

if(s.compareTo(s1)>0)

p("compareTo:"+"True");

else

p("False");

p("indexOf():"+s.indexOf('P'));

p("lastIndexOf():"+s.lastIndexOf('a'));

p("substring():"+s.substring(7,15));

String s3="Dheeraj baweja";

p("concat():"+s3.concat(s2));

p("replace():"+s3.replace('D','d'));

p("trim():"+s3.trim());

}

public void BufferOperation()

{

StringBuffer buf=new StringBuffer(s);//capacity of the created buffer.

buf.ensureCapacity(400);//specify the capacity of buf.

p("capacity():"+buf.capacity());//specify the capacity of any object.

buf.setLength(100);

p("setLength():"+buf.length());

buf.setCharAt(7,'p');

p("setCharAt():"+buf);

buf.append(" Varanasi");

p("append():"+buf);

buf.insert(19,"Sarnath");

p("insert():"+buf);

buf.reverse();

p("reverse():"+buf);

buf.reverse();

buf.delete(19,26);

p("delete():"+buf);

buf.deleteCharAt(7);

p("deleteCharAt():"+buf);

buf.replace(7,12,"Prasad");

p("replace():"+buf);

}

}

class mystring

{

public static void main(String args[])

{

System.out.println("\t\tWelocme to the java program.");

str ob=new str();

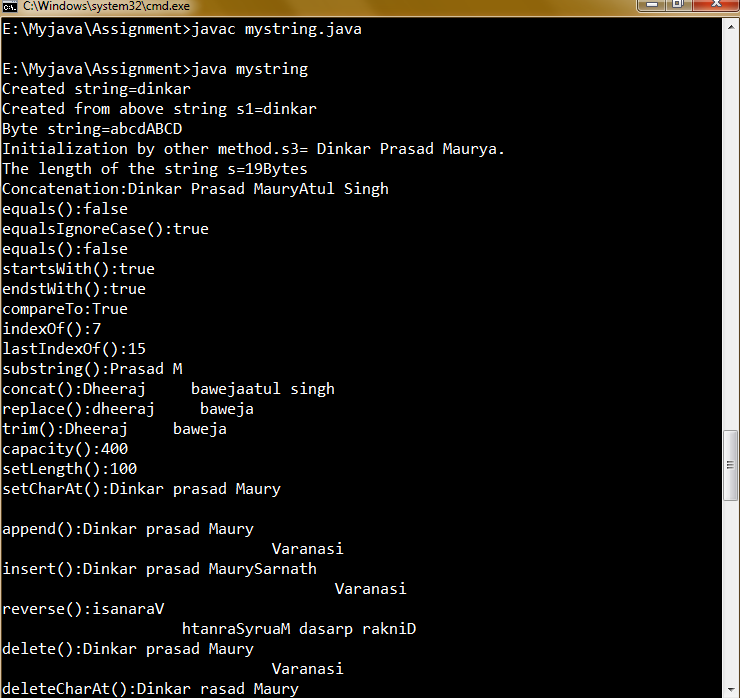
ob.operation();

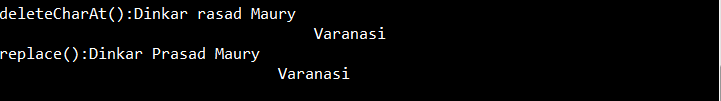
ob.BufferOperation();

}

}

## OUTPUT:-





## PROGRAM

## //Program to create a class to implement ByteStream in java.

BYTE INPUT STREAM:-

import java.io.\*;

class FileInputStreamDemo

{

public static void main(String args[])throws IOException

{

System.out.println("\t\tWelcome to the java program.");

int size;

InputStream f=new FileInputStream("FileInputStreamDemo.java");

System.out.println("Totle available Bytes:"+(size=f.available()));

int n=size/40;

System.out.println("First "+n+"bytes of the file on read() at a time");

for(int i=0;i<n;i++)

System.out.print((char)f.read());

System.out.println("\nStill available:"+f.available());

System.out.println("Reading the next "+n+" with one read(b[])");

byte b[]=new byte[n];

if(f.read(b)!=n)

System.out.println("Could not read "+n+"Bytes.");

System.out.println(new String(b,0,n));

System.out.println("\nStill available:"+f.available());

System.out.println("Skipping half of the remaning bytes with skip()");

f.skip(size/2);

System.out.println("Still available" +f.available());

System.out.println("Reading "+n/2+" in to the end of the array.");

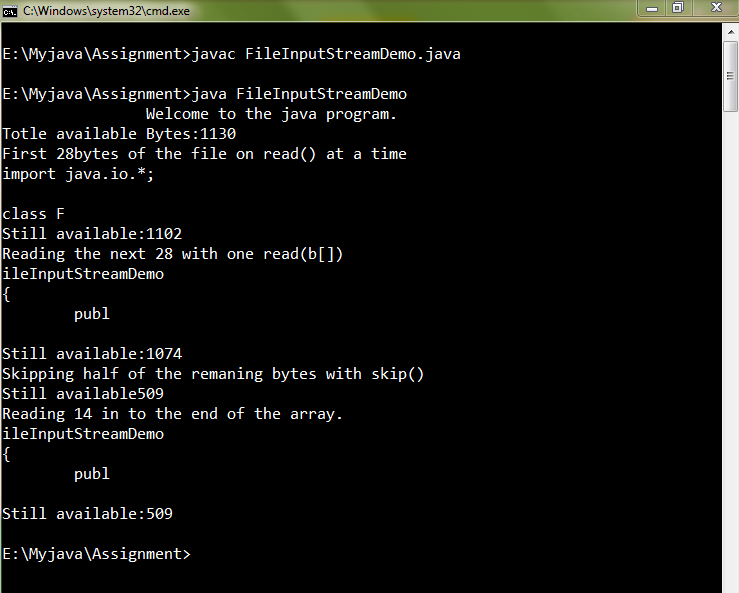
System.out.println(new String(b,0,b.length));

System.out.println("\nStill available:"+f.available());

f.close();

} }

## OUTPUT:-



BYTE OUTPUT STREAM:-

import java.io.\*;

class FileOutputStreamDemo

{

public static void main(String args[])throws IOException

{

System.out.println("\t\tWelcome to the java program.");

String source="Dinkar Prasad Maurya\n"+"Atul singh"+"Deepanker";

byte buf[]=source.getBytes();

OutputStream f0=new FileOutputStream("file1.txt");

for(int i=0;i<buf.length;i++)

f0.write(buf[i]);

f0.close();

OutputStream f1=new FileOutputStream("file2.txt");

f1.write(buf);

f1.close();

OutputStream f2=new FileOutputStream("file3.txt");

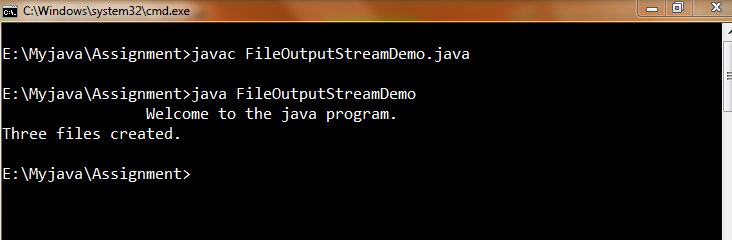
f2.write(buf,buf.length-buf.length/4,buf.length/4);

f2.close();

}

}

## OUTPUT:-



Created Files:-



The content of the files are:-

File1:- Dinkar Prasad MauryaAtul singhDeepanker

File2:- Dinkar Prasad MauryaAtul singhDeepanker

File3:- hDeepanker

## PROGRAM

## //Program to create a class to implement CharacterStream in java.

CHARACTER READER:-

import java.io.\*;

class FileReaderDemo

{

public static void main(String args[])throws IOException

{

System.out.println("\t\tWelcome to the java program.");

FileReader fr=new FileReader("FileReaderDemo.java");

BufferedReader br=new BufferedReader(fr);

String s;

while((s=br.readLine())!=null)

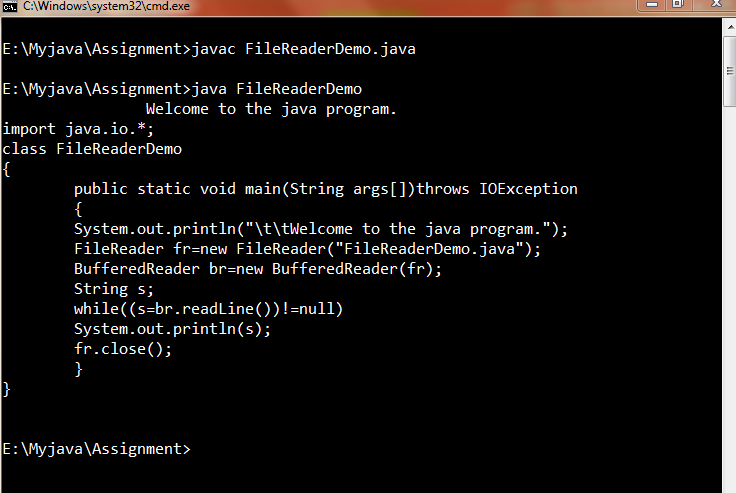
System.out.println(s);

fr.close();

}

}

## OUTPUT:-



CHARACTER WRITER:-

import java.io.\*;class FileWriterDemo

{

public static void main(String args[])throws IOException

{

System.out.println("\t\tWelcome to the java program.");

String source="Dinkar Prasad Maurya\n"+"Atul singh"+"Deepanker";

char buf[]=new char[source.length()];

source.getChars(0,source.length(),buf,0);

FileWriter f0=new FileWriter("file4.txt");

for(int i=0;i<buf.length;i+=2)

f0.write(buf[i]);

f0.close();

FileWriter f1=new FileWriter("file5.txt");

f1.write(buf);

f1.close();

FileWriter f2=new FileWriter("file6.txt");

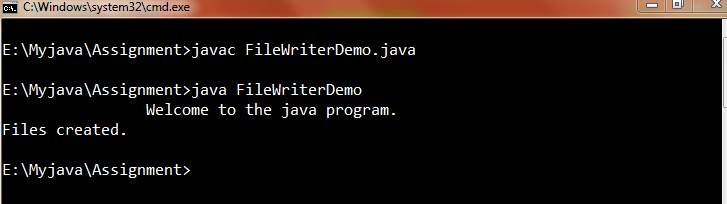
f2.write(buf,buf.length-buf.length/4,buf.length/4);

f2.close();

System.out.println("Files created.");

}}

## OUTPUT:-



Created Files:-



The content of the files are:-

File4:- Dna rsdMuytlsnhepne

File5:- Dinkar Prasad MauryaAtul singhDeepanker

File6:- hDeepanker

## PROGRAM

## //Program to create a simple applet in java.

import java.applet.\*;

import java.awt.\*;

/\* <applet code="myapplet" width=300 height=300>

</applet>\*/

public class myapplet extends Applet

{

String msg="Welcome to the java program.";

String ap="A simple applet created.";

String s="Pankj Dinkar";

//Called once during the runtime of applet.

public void init()

{

s+="Prasad";

}

//Called each time an appplet'sHTML document displayed on screen.

//Also called to restart the applet.

public void start()

{

s+="Maurya";

}

public void destroy()//Called by the applet browser just before applet is terminated.

{

s+="Sarnath";

}

public void stop()//Suspned the execution of the applet.

{

s+="Varanasi";

}

//When applet begins execution.

public void paint(Graphics g)

{

g.drawString(msg,20,50);

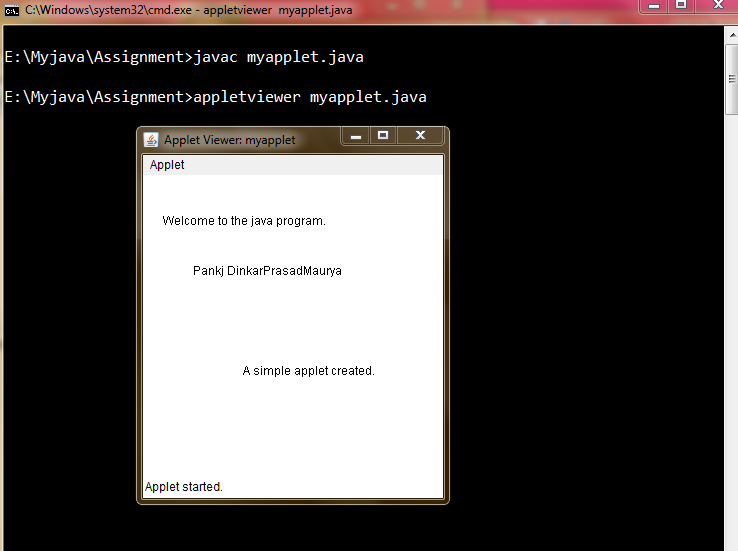
g.drawString(s,50,100);

g.drawString(ap,100,200);

}

}

## OUTPUT:-



## PROGRAM

## //Program to create a socket in java.

//CLIENT SOCKET

import java.net.\*;

import java.io.\*;

public class Client

{

public static void main(String args[])

{

System.out.println("\t\tWelcome to the java program.");

int serverPort = 6666; //Give the port number on which the server is listening.

String address = "192.168.1.5"; // this is the IP address of the server program's computer.

// the address given here means "the same computer as the client".

try

{

// create an object that represents the above IP address.

InetAddress ipAddress = InetAddress.getByName(address);

System.out.println("Any of you heard of a socket with IP address " + address + " and port " + serverPort + "?");

// create a socket with the server's IP address and server's port.

Socket socket = new Socket(ipAddress, serverPort);

System.out.println("Yes! I just got hold of the program.");

// Get the input and output streams of the socket, so that you can receive and send data to the client.

InputStream sin = socket.getInputStream();

OutputStream sout = socket.getOutputStream();

// Just converting them to different streams, so that string handling becomes easier.

DataInputStream in = new DataInputStream(sin);

DataOutputStream out = new DataOutputStream(sout);

// Create a stream to read from the keyboard.

BufferedReader keyboard = new BufferedReader(new InputStreamReader(System.in));

String line = null;

System.out.println("Type in something and press enter. Will send it to the server and tell ya what it thinks.");

System.out.println();

while(true)

{

line = keyboard.readLine(); // wait for the user to type in something and press enter.

System.out.println("Sending this line to the server...");

out.writeUTF(line); // send the above line to the server.

out.flush(); // flush the stream to ensure that the data reaches the other end.

line = in.readUTF(); // wait for the server to send a line of text.

System.out.println("The server was very polite. It sent me this : " + line);

System.out.println("Looks like the server is pleased with us. Go ahead and enter more lines.");

System.out.println();

}

}

catch(Exception x)

{

x.printStackTrace();

}

}

}

SERVER SOCKET:-

import java.net.\*;

import java.io.\*;

public class Server

{

public static void main(String[] ar)

{

System.out.println("\t\tWelcome to the java program.");

int port = 6666; // just a random port.It must lie between 1025 and 65535.

try

{

ServerSocket ss = new ServerSocket(port); // create a server socket and bind it to the above port number.

System.out.println("Waiting for a client...");

Socket socket = ss.accept(); // make the server listen for a connection, and let you know when it gets one.

System.out.println("Got a client :) ... Finally, someone saw me through all the cover!");

System.out.println();

// Get the input and output streams of the socket, so that you can receive and send data to the client.

InputStream sin = socket.getInputStream();

OutputStream sout = socket.getOutputStream();

// Just converting them to different streams, so that string handling becomes easier.

DataInputStream in = new DataInputStream(sin);

DataOutputStream out = new DataOutputStream(sout);

String line = null;

while(true)

{

line = in.readUTF(); // wait for the client to send a line of text.

System.out.println("The dumb client just sent me this line : " + line);

System.out.println("I'm sending it back...");

out.writeUTF(line); // send the same line back to the client.

out.flush(); // flush the stream to ensure that the data reaches the other end.

System.out.println("Waiting for the next line...");

System.out.println();

}

}

catch(Exception x)

{

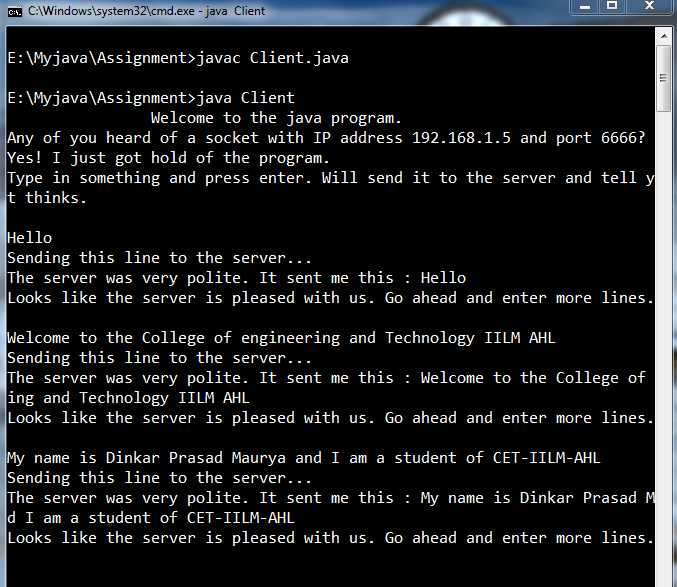
x.printStackTrace();

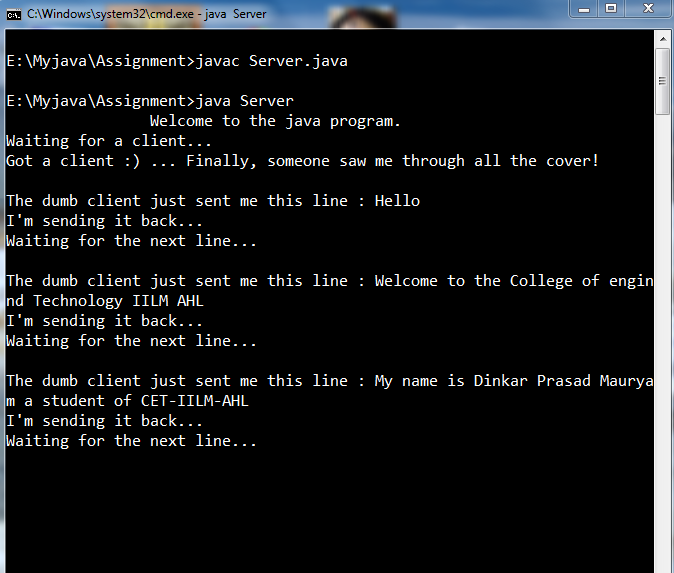
}

}

}

## OUTPUT:-





## PROGRAM

## //Program to create a listener class in java.

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*<applet code="Listener" width=300 height=300>

</applet>\*/

public class Listener extends Applet implements MouseListener,MouseMotionListener

{

String msg="";

int mouseX=10,mouseY=40,movX=0,movY=0;

TextField t1;

public void init()

{

addMouseListener(this);

addMouseMotionListener(this);

}

//mouse listener

public void mouseClicked(MouseEvent me)

{

showStatus("Mouse clicked.");repaint();

}

public void mouseEntered(MouseEvent me)

{

showStatus("Mouse Entered.");

repaint();

}

public void mouseExited(MouseEvent me){}

public void mousePressed(MouseEvent me){}

public void mouseReleased(MouseEvent me){}

//mouse motion listener.

public void mouseDragged(MouseEvent e)

{

mouseX=e.getX();mouseY=e.getY();

movX=e.getX();movY=e.getY();

msg="\*"; repaint();

}

public void mouseMoved(MouseEvent e)

{

movX=e.getX();movY=e.getY();

repaint();

}

public void paint(Graphics g)

{

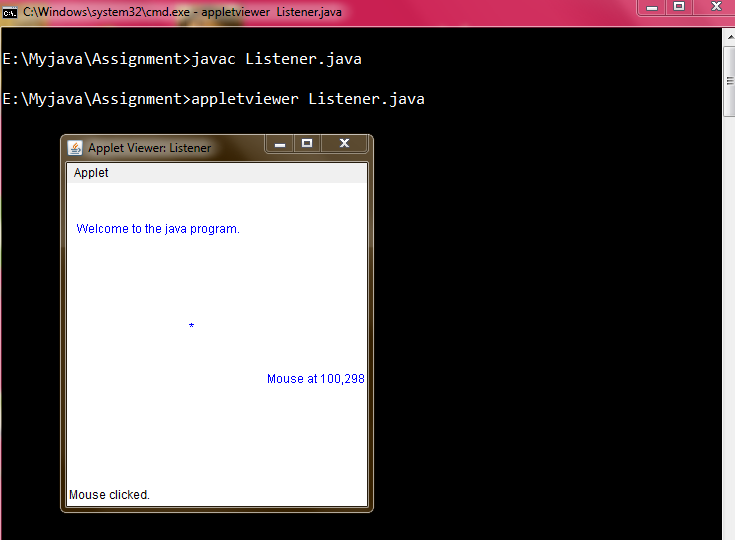
g.setColor(Color.blue);

g.drawString("Welcome to the java program.",10,50);

g.drawString(msg,mouseX,mouseY);

g.drawString("Mouse at "+movX+","+movY,200,200);}}

## OUTPUT:-



## PROGRAM

## //Program to create a AWT panel in java.

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*<applet code="awtpanel" width=300 height=300>

</applet>\*/

public class awtpanel extends Applet implements ActionListener

{

int a,b,c=0;

TextField t1,t2,t3;

Button b1,b2;

public void init()

{

setLayout(new GridLayout(5,1));

t1=new TextField();

add(t1);//add the text box to the window.

t2=new TextField();

add(t2);//add the text box to the window.

t3=new TextField();

add(t3);//add the text box to the window.

b1=new Button("\*");

add(b1);

b1.addActionListener(this);

b2=new Button("Result");

add(b2);

b2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b1)

{

a=Integer.parseInt(t1.getText());

b=Integer.parseInt(t2.getText());

c=a\*b;

}

if(e.getSource()==b2)

{

if(c>0)

t3.setText(String.valueOf(c));

else

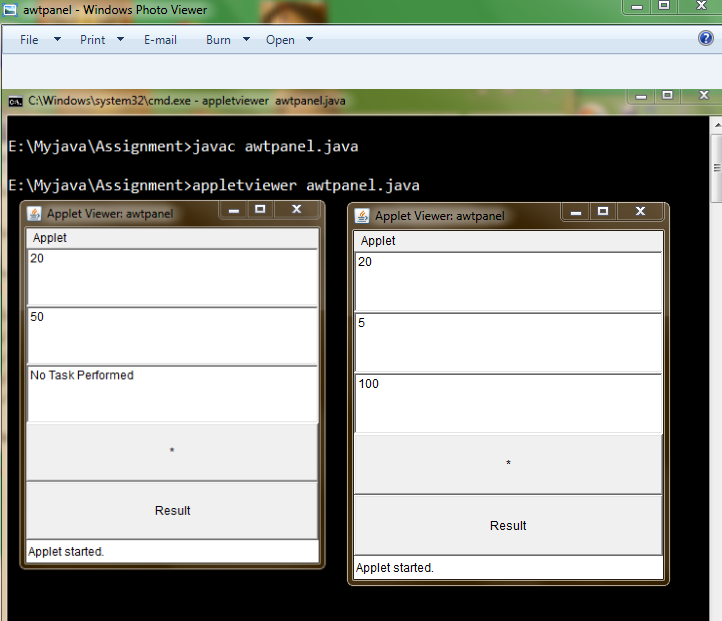
t3.setText("No Task Performed");

}

}

}

## OUTPUT:-



## PROGRAM

## //Program to show a grid layout in java.

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*<applet code="Grid2" width=300 height=300>

</applet>\*/

public class Grid2 extends Applet

{

static final int n=3;

public void init()

{

setLayout(new GridLayout(2,n));//row\*column.

setFont(new Font("SansSerif",Font.BOLD,30));

int k=65;

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

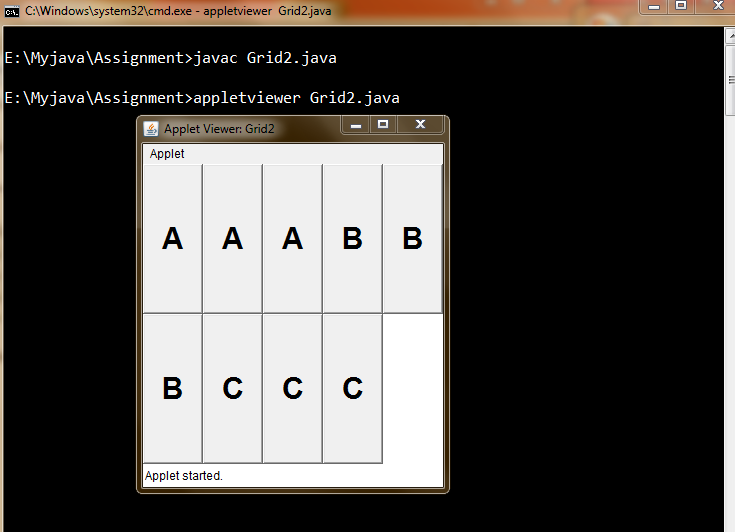
add(new Button(""+(char)k));

}

k++;

} } }

## OUTPUT:-



## PROGRAM

## //Program to show a flow layout in java.

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*<applet code="Flow" width=300 height=300>

</applet>\*/

public class Flow extends Applet implements ItemListener

{

String msg="";

Checkbox c1,c2,c3;

public void init()

{

//check boxes are arrenged in linear seqence.

setLayout(new FlowLayout(FlowLayout.LEFT));

c1=new Checkbox("IILM",null,true);

add(c1);

c1.addItemListener(this);

c2=new Checkbox("NIET");

add(c2);

c2.addItemListener(this);

c3=new Checkbox("IEC",null,false);

add(c3);

c3.addItemListener(this);

}

public void itemStateChanged(ItemEvent e)

{

repaint();

}

public void paint(Graphics g)

{

g.drawString("Welcome to the java program.",50,50);

msg="Curren State:";

g.drawString(msg,20,100);

msg=" IILM"+c1.getState();

g.drawString(msg,20,150);

msg=" NIET"+c2.getState();

g.drawString(msg,20,200);

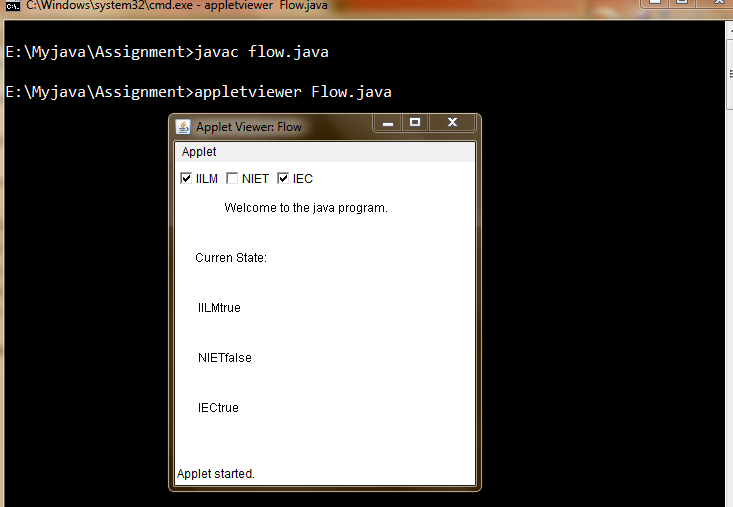
msg=" IEC"+c3.getState();

g.drawString(msg,20,250);

}

}

## OUTPUT:-



## PROGRAM

## //Program to create a JDBC-ODBC connectivity in java

import java.sql.\*;

public class myodbc

{

public static void main(String args[])

{

System.out.println("\t\tWelcome to the java program.");

System.out.println("The program for odbc connectivity is executing.");

Connection con;

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:myodbc");

Statement stmt=con.createStatement();

stmt.executeUpdate("UPDATE Student"+ " SET Name='Dheeraj Baweja'"+ " WHERE Roll=0815010031");

stmt.close();

con.close();

}

catch(ClassNotFoundException e)

{

System.out.println(e); }

catch(SQLException e1)

{

System.out.println(e1);

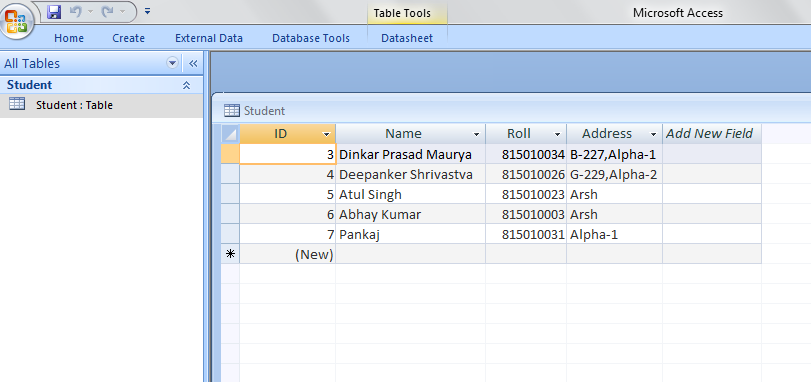
}

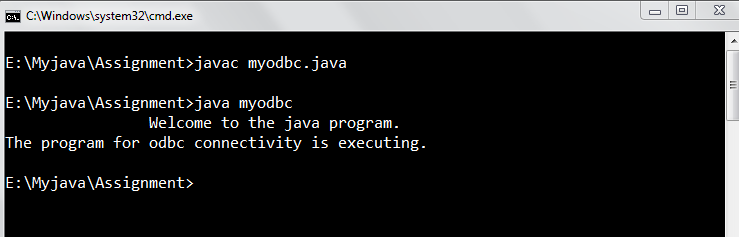
}

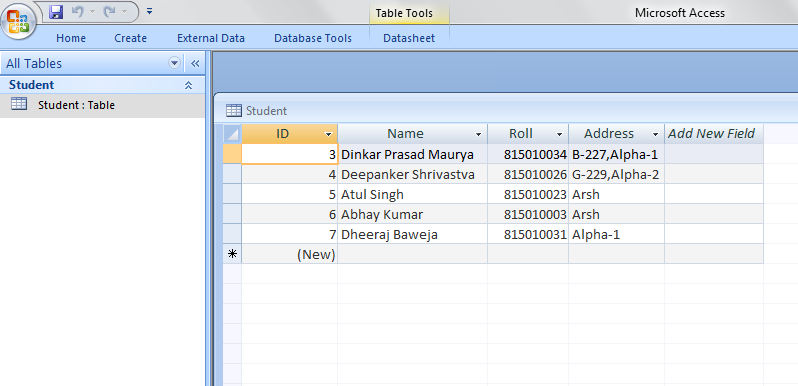
}

## OUTPUT:-









## PROGRAM

## //Program to create a Swing applet in java.

import java.awt.\*;

import javax.swing.\*;

/\*<applet code="swingapplet" width=300 height=300>

</applet>\*/

public class swingapplet extends JApplet

{

public void init()

{

Container contentPane=getContentPane();

ImageIcon ii=new ImageIcon();

//Create a label.

JLabel j1=new JLabel("Welcome to the java program",ii,JLabel.CENTER);

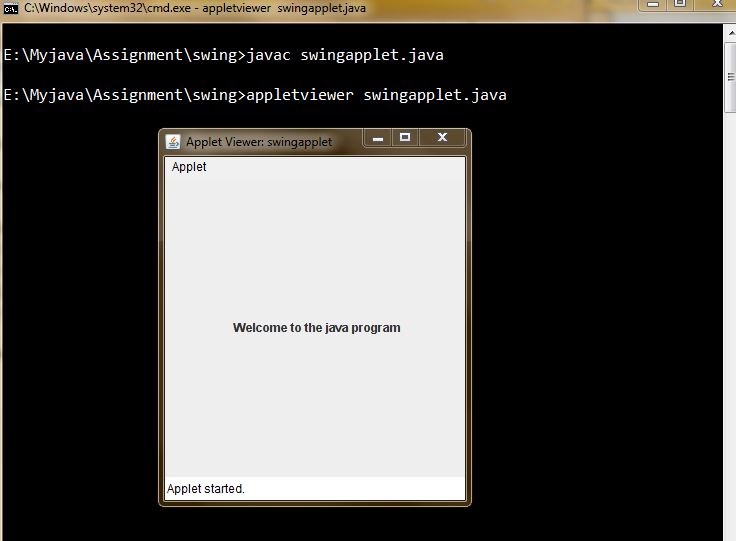
//Add label to the content pane.

contentPane.add(j1);

}

}

## OUTPUT:-



## PROGRAM

## //Program to create a panel in Swing in java.

import java.awt.\*;

import javax.swing.\*;

/\*<applet code="swingpanel" width=300 height=300>

</applet>\*/

public class swingpanel extends JApplet

{

JTextField t1;

public void init()

{

Container contentPane=getContentPane();

contentPane.setLayout(new FlowLayout());//It is a panel.

//add text field field to the content pane.

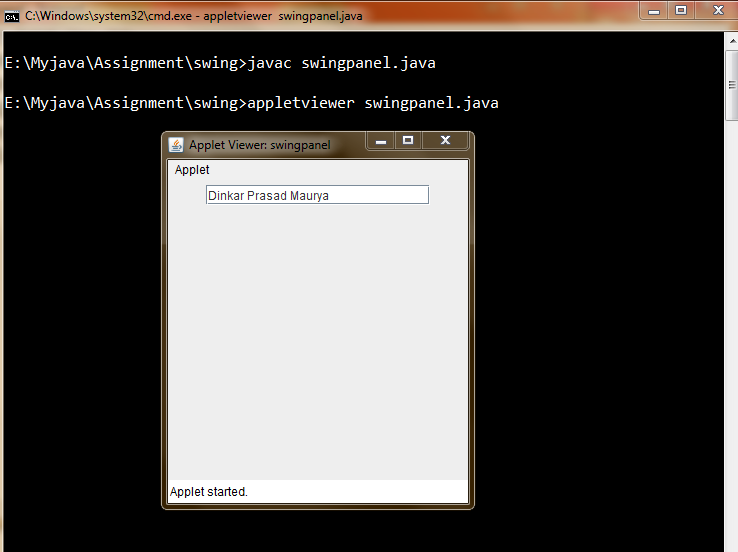
t1=new JTextField(20);

contentPane.add(t1);

}

}

## OUTPUT:-



## PROGRAM

## //Program to create an event in Swing in java.

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

/\*<applet code="Swingevent" width=300 height=300>

</applet>\*/

public class Swingevent extends JApplet implements ActionListener

{

JTextField t,t1;

public void init()

{

Container contentPane=getContentPane();

contentPane.setLayout(new FlowLayout());

ImageIcon atul =new ImageIcon();

JButton b1=new JButton(atul);

b1.setActionCommand("Atul Singh");

b1.addActionListener(this);

contentPane.add(b1);

ImageIcon Dinkar =new ImageIcon();

b1=new JButton(Dinkar);

b1.setActionCommand("Dinkar Prasad Maurya");

b1.addActionListener(this);

contentPane.add(b1);

ImageIcon dheeraj =new ImageIcon();

b1=new JButton(dheeraj);

b1.setActionCommand("Dheeraj Baweja");

b1.addActionListener(this);

contentPane.add(b1);

ImageIcon deepanker =new ImageIcon();

b1=new JButton(deepanker);

b1.setActionCommand("Deepanker Shrivastava");

b1.addActionListener(this);

contentPane.add(b1);

t1=new JTextField(15);

contentPane.add(t1);

t=new JTextField(15);

contentPane.add(t);

}

public void actionPerformed(ActionEvent e)

{

t1.setText(e.getActionCommand());

t.setText("Welcome to the java program."); }}

## OUTPUT:-

